AD-A095 289 FOREIGN TECHNOLOGY DIV WRIGHT-PATTERSON AFB OH F/G 17/2
NEW APPROACH TO THE SOFTWARE DEVELOPMENT OF THE STORED PROGRAM --ETC(U)
DEC 80 M MEKINDA
UNCLASSIFIED FTD-ID(RS)T-1152-80 NL

1 OF 1
AD
A095289

END
DATE
FILMED
3 81
DTIC

FTD-ID(RS)T-1152-80

2

15

# FOREIGN TECHNOLOGY DIVISION

AIR FORCE SYSTEMS COMMAND

NEW APPROACH TO THE SOFTWARE DEVELOPMENT OF THE STORED PROGRAM TELEPHONE EXCHANGES

by

Milan Mekinda

FOREIGN TECHNOLOGY DIVISION

Approved for public release;
distribution unlimited.

FTD-ID(RS)T-1152-80

# EDITED TRANSLATION

/ // FTD-ID(RS)T-1152-80 // 10 Dec[redacted]980

MICROFICHE NR: FTD-80-C-001223

NEW APPROACH TO THE SOFTWARE DEVELOPMENT OF THE
STORED PROGRAM TELEPHONE EXCHANGES

By: Milan/Mekinda

English pages: ~~12~~

~~Source:~~ Elektrotehniski Vestnik, Nr. 1, 1977,
pp. 53-56

Country of origin: (Yugoslavia)
Translated by: LINGUISTIC SYSTEMS, INC.
F33657-78-D-0618
Nenad Popovic
Requester: FTD/TQCS
Approved for public release; distribution unlimited.

Accession For
NTIS GRA&I ☒
DTIC TAB □
Unannounced □
Justification

By
Distribution/
Availability Codes

| Dist | Avail and/or Special |
|---|---|
| A | |

THIS TRANSLATION IS A RENDITION OF THE ORIGINAL FOREIGN TEXT WITHOUT ANY ANALYTICAL OR EDITORIAL COMMENT. STATEMENTS OR THEORIES ADVOCATED OR IMPLIED ARE THOSE OF THE SOURCE AND DO NOT NECESSARILY REFLECT THE POSITION OR OPINION OF THE FOREIGN TECHNOLOGY DIVISION.

PREPARED BY:

TRANSLATION DIVISION
FOREIGN TECHNOLOGY DIVISION
WP.AFB, OHIO.

14

Iskra Technical Reports
1977 - 1

# New Approach to the Software Development of the Stored Program Telephone Exchanges

by

Milan Mekinda,Kranj

UDK 681.3.06:621.395.722 (Received:Jan.20,1977)

(Novi prijemi razvoja programske opreme programsko vodenih telefonskih centrala*)

Stored program control systems have reached an important triumph after entering the telecommunication field,

Introduction of computers-processors in telecommunication systems offers a new quality and enables an enormous development.

Outstanding tasks of the designers of the SPC exchanges are shown. Solutions of the problems which enable an even faster development of systems are outlined.

In this report the most significant features of the SPC designed telephone exchanges are surveyed. Those solutions to problems which will lead to even faster further growth are shown here.

Constant study of advances in programming science and the introduction of the new knowledge about the SPC systems have already gained confidence in the SPC exchanges, which will continue to he asserted also in the future.

* Report submitted to the 10th Yugoslav Symposium on Telecommunications, Lyublyana, Oct.5-6,1976.

## Issues related to the development of stored program control systems

Stored program control (SPC) systems have attained their first important triumph after entering into the telecommunications field. They have demonstrated that the introduction of computers into the corresponding programs and data processors has given the telecommunication systems a new qualitative dimension which has paved the way for their great development. SPC systems have gained the justified confidence of their customers and manufacturers. This new quality has, indeed, added to the SPC systems many new features. We can say that the introduction of computers into telecommunication systems actually represents a new revolution in their development.

It is no longer necessary to try to convince manufacturers of the great advantages of the SPC systems. Having gotten to know them, the users also manifest a continuous interest for them. There is also a general agreement that SPC systems represent the right road for the quick further development of telecommunications. Many different stored program controlled systems in various branches of telecommunications are now being operated all over the world.Everyone who does not want to miss the express train of telecommunications development knows that they must promptly become closely acquainted with the SPC system in order to start availing themselves of the SPC's outstanding features.

Thus, the SPC system has already won the first great battle for its affirmation. This system has already proven itself to be safe and failproof. Nevertheless, this system now has to face another and even more difficult struggle since interest for this system in the world has grown enormously. The manufacturers of the SPC system also face important challenges now. They must further develop the SPC system and improve their productivity in manufacturing in order to satisfy their customers. The public is aware of great abilities of manufacturers and therefore expects even greater achievements from them. Future orders for SPC systems will relate to the following areas:

- capabilities,
- adaptability,
- reliability, and
- documentation.

The influence of those who utilize the systems on manufacturers in the software field will be increasing. As the users become more acquainted with the SPC system and learn to know more about them, they will be exerting much greater influence. This is a normal process since the users themselves are those who will want to use the new facilities to the best their advantage. Therefore the users will increasingly insist, for instance, on more simplified preparation of the processing functions, which now exist within the new system. Consequently, both developers and manufacturers will have to adjust themselves and to correspondingly modify software development in order to meet requirements. This means that the software engineering will have to be easily adaptable and well documented.

Telecommunications indeed have developed very quickly. The impulse coded modulated (PCM) transmission of voice and data first appeared. Integrated systems are already knocking on the door and those who use them want the soonest possible application of the newly developed facilities. At the present, it is important to mention that SPC system developers need to shorten the time necessary for the development,production and - most of all - for new connections. This means that the price of products has to be reduced and that competition among the manufacturers will have to intensify.

The national interests require a faster pace of the telecommunications development since the telecommunications are an important part of the entire national overhead.For this reason a greater part of the national investment allocations should be given to the telecommunications. A fast growth of telecommunications causes both telecommunications users and manufacturers to interact and support further relecommunications development.

This is the general environment which influences the developers of SPC systems. A guaranteed increase in the productivity of the corresponding software program engineering is needed. Without an increase in productivity, we will not be able to satisfy the need for lower costs, nor will we be able to insure an output of the necessary high quality and reliability of products and services.

## Productivity in software engineering for SPC systems.

When conducting a productivity evaluation, one should not only consider the length of development time and its cost, but also the amount of labor time which was used as well as the cost of production. All of this includes tests, the installation and putting into operation, and the maintenance. The most significant factors which determine the productivity are the following:

- supervising methods of software engineering,
- personnel participating in the programming,
- the project's concept and scope,
- the production and test facilities.

The software engineering of the SPC systems is highly complex. A great number of people participate in its development, manufacturing, and putting the system into operation. For this reason it is very important to establish a proper work organization of the project, as well as the personnel deployment into working teams. It is necessary to set a proper and clear direction and management by integrating various teams which are involved in different project stages. The task of directing a software engineering project is quite specific. Thus, for instance, principles by which other projects are conducted, are not applicable here. New management techniques are necessary, and they have to be fully adjusted to the specific character of the software engineering.

Methods relating to structural programming and to the use of various team patterns are known. However, it is difficult to follow them. For a work at programming it is important to keep in mind that the job of developing electronic circuits is always a kind of creative artistic process. The programming is conducive to more work freedom and to easier unrestrained expression of ideas. Programming work motivates imaginative persons. This kind of people fit into the required discipline of routine rules and standards with difficulty. Such a situation is, indeed, even more difficult for team leaders. Therefore it is useful to apply broader program formulations. It is also useful to introduce standard macro-guidelines in subprogramming work as well as to utilize a higher level of programming and problem oriented computer languages. Under such conditions the creative abilities of the programmers can become an instrument towards achieving an optimal work organization, in addition to applying standardized elements.

In managing software engineering projects, it is a great mistake to underestimate the amount of work on material equipment. This usually happens when team leaders themselves lack the necessary qualifications for software engineering. Even team leaders sometimes think that programming is not scientific work and that every bright person can do it. Team leaders of this kind will never be successful in their programming activities.

An important problem for engineering leaders represents the fact that the field of principles for programming planning is not yet sufficiently developed, or if it is, then it is not enough or suitable. For instance, suppose a programmer has performed software engineering work for which he spent one year. It would be a mistake to assume that same work could be done by four programmers in three months. These two things do not represent "one programming year". Or, at least, in a majority of cases this would not be true.Everything depends on the nature of the work, on the extent to which the work can be divided into separate components, on the knowledge of and experience with subprograms, as well as on the correct evaluation of the mutual interdependence of different working teams. The best way of planning is to select a suitable number of experienced and reliable experts, whose knowledge is updated, and then to let them evaluate candidates for such an assignment. Thus, the task of planning is to mate particular kind of work with the abilities of programmers.

The next point in the evaluation of software engineering work is to pay great attention to the number of set printed programming instructions. It is clear that this is both an unreliable and unjustifiable criterion for the measurement of the programmers' productivity. A great deal depends on the complexity of the problems and on the techniques of a given software engineering task. It is somewhat better to measure the programmers' time used for the programming and testing. This kind of evaluation needs team leaders with high programming knowledge and on an outstanding understanding of the job.

It is a difficult and responsible duty of a programming team-leader to prepare a plan which would be both realistic and stimulating. Such a plan should influence programmers and induce in them a constructive attitude. Unrealistic plans are debililitating since, even in advance, they reduce work productivity and are doomed to failure.

Productivity depends to a great extent on the participating personnel. For the software engineering of SPC systems the most promising personnel are those who possess a mixture of technical and programming knowledge. It is, however, usual that personnel involved have either technical or programming expertise, but rarely both.

Productivity depends substantially on applied organizational concepts, as well as on the kind of programming equipment. If we want to optimize the realization of processing functions by means of suitable work organization, it may easily happen that such an approach is neither sufficiently flexible nor is it amenable to accept innovating functions, or the other way round. In the course of the development of software engineering, we have to prepare a simple and easy design which will be suitable for any later adaptations. What really has to be done is to design a structure which will have the greatest possible flexibility. This allows for the use of already known designing techniques, such as:

- a hierarchical set-up from top to bottom,
- a structural programming approach, and
- to proceed step-by-step.

The approach to designing of the software engineering can be either procedural (how), or functional (what).

A great number of processing functions are identical to various SPC systems. Therefore it is necessary to endeavor to standardize separate process functions in order to extend the application of accepted solutions. The first step towards this goal is to standardize the communications between man and machine. This means the use of a uniform common language for man-machine communication. Also, the work of the users will be much easier and simpler, since different SPC systems already to exist. This would, of course, also represent an even greater advantage for manufacturers who will have to clearly define such a computer language and to make it accessible and understandable to all. The standardization does not only improve productivity, but it, in addition, substantially reduces the time needed to learn and to get along with systems.

Factors which also influence the productivity very much, are to be found in various facilities for an easy and simplified analysis, the programming, inspecting and servicing of the SPC systems. Such facilities are:

- A standard macro-guide for subprograms which helps the programmer in his work and increases the reliability of software engineering.

- Program inspections which simplify and increase efficiency by applying the testing programs. As the availability of such programs increases, work on uncovering software engineering errors becomes more successful. Assuming that we have saved 50% of the time needed for the development, manufacturing and inclusion of the programs, it is evident how important it is to cut down the testing time.

- An ample and rich operational system of the computer center which makes possible a higher degree of automatization during the equipment manufacturing.

- Programs for the automatic output of documents (flow diagrams, function surveys, etc.)

It is necessary to pay greater attention to all these facilities because they strongly influence productivity, reduce costs and make the work easier.

## The adaptability of the SPC systems.

The adaptability of the SPC systems can be easily recognized both by the processors and the manufacturers. A processor expects ,first, such an adaptability which will be manifested in simple and efficient utilizization of the system. He is looking for a simple way how to update those data which are relevant for the process functions (such as customers data, billing, directions, preparedness,etc). The processors want to to be able to efficiently prepare the system for operation (statistical data, observations, monitoring). They want to have a routine way of testing operations and reliable method of billing. In the future, we may well expect that processors will want to have the following available: statistical data, monitoring the system, alternative interpretations and analyses of the data, new routine tests and checks, and additional processing of billing data. A SPC system, however, must be constructed in such manner that all these supporting (administrative) functions do not repeatedly interfere with the operational processing functions.Programs of supporting functions contain autonomous program modules, which are mutually independent.

From the point of view of a manufacturer, the idea of adaptability is much broader. An adaptable SPC system is one which enables the simple realization of a great number of different applications. The software must be reliably engineered so that it offers the desired adaptability. Full adaptability is possible only when the modular structure is based on an analysis of the entire system structure. Modular software equipment consists of a great number of elementary programs-modules, which are mutually integrated according to the rules which are set down in the corresponding reliable tables. The entire logic of process functions is set down in such tables. Elementary modules do not require separate processing logic.

Only the elementary actions are separated. When necessary, new elementary modules are added. The most important thing is that all of this does not require any extra programming.

The flexible organization of software engineering for SPC systems offers a simple and quick adaptation to various circumstances. The applicability of a system consists in avoiding to do new programming as much as possible, and thus to save the time which would be needed for testing. Namely, a new application only requires that the newly introduced modules have to be tested.

Specific issues connected with software engineering.

One of more important problems is the "invisibility" of software engineering. We cannot see the software at work. What can be observed is not sufficient for arriving at any conclusions about their adequacy. Therefore, it is necessary to find some way of visualizing software operations. A partial visualization is possible at the computer control desk.This, of course, is not enough and consequently there are necessary other means. Through software engineering we build-in separate tested programs, which are actuated according to need. Such a program can be reliably used to achieve a visual observation of software performance, since it is printed on the incoming-outgoing device. This is how a program and its piint-out may supply structural data.

Another issue is that of the complexity of the software engineering. Tens of thousands of instructions and an extended data structure may easily be contained there. The production of such systems requires tens of programmer/years. Documentation which would supply a visual representation of such a system, could reach a volume of thousands of pages. As long as the designers of software engineering do not see or understand the system structure, there is little hope that they could be able to effect a qualitative system modification.

A good feature of software engineering is its integrity. When such a system is once finished and tested, it then acts without failures up to the end of its useful life. This feature should be maximally utilized.

Testing of SPC systems software engineering.

The testing of software causes usually great difficulties in case of large systems which are turned on at fixed time points. Delays on the account of testing are frequent and long. At the moment this item represents the most serious problem in regard to the SPC systems. It is imperative to research for new methods of testing, since the presently applied methods are not satisfacory.

The opinion that the most appropriate time to begin the program testing is when such systems are readied for operations is totally wrong. The testing ought to be started as early as when the specifications for the software engineering are about to be prepared. From this point on during the entire course of the software engineering - namely the analysis and the formulation of the software specifications - is also the period of testing. The testing goes and continues concurrently with the setting up of programs all the way through up to the moment when all these programs are put into operation. Thus, the main part of the software engineering must be designed and applied in such a manner that the testing could be conducted and performed during all the stages of its development. This is possible thanks to the modular structure, which enables step-by-step construction as concurrently are being executed tests with the software equipment. First of all, the controlling programs of software are tested as are set their standard circuits. This represents the basic scheme, which is easy to perform and takes little time. We then proceed to develop individual elementary modules. As they are finished, they are added to the basic structure. Each added module is being tested, both separate and within the entire system as its integral part. Thus, we gradually build up the software engineering and test it as we go. When the last module is added and tested, the entire software engineering is finished and ready for operation. As the software engineering is formed step-by-step, we also establish the relevant documentation.

## Conclusion

In this report the most important tasks of the SPC system developers are presented. Solutions to some problems which appear on the account of the need for the fastest possible system development, are also shown here. Also, some advice to designers of software engineering are given, in order to make their work easier and more successful, which could improve some features of the SPC systems. The constant scientific progress in the field of programming and the adoption of new breakthroughs in SPC systems, confirm the general confidence in their increasing feasibility and importance.

### • Literatura

[1] *D. E. Knuth:* Computer Programming as an Art ACM 1974

[2] Conference on Software Engineering for Telecommunication Switching Systems Essex 1973

[3] Conference on Software Engineering for Telecommunication Switching Systems Salzbung 1976

[4] *M. Mekinda:* »Studij in raziskava programske opreme lokalne centrale sistema METACONTA 10 C in možnost optimizacij«. Naloga 1837 sklada Boris Kidrič, Ljubljana 1975

[5] *M. Mekinda:* »Programsko vodeni sistem telefonske centrale«. Magistrska naloga, Fakulteta za elektrotehniko, Ljubljana 1974.

Avtorjev naslov: Mag. Milan **Mekinda**, dipl. ing., Iskra — Telekomunikacije, Kranj

FILMED

-8